Слайд 1

Титульный лист

Слайд 2

UDP является протоколом транспортного уровня (4 уровень модели OSI)

Задачей транспортного уровня является доставка данных

Слайд 3

Протокол пользовательских дейтаграмм (User Datagram Protocol) выполняет задачи транспортного уровня модели OSI , является альтернативой TCP.

Обычно применяется поверх протокола IP.

При этом в заголовке IP указывается номер вышележащего протокола 17

Слайд 4

Особенностью протокола UDP является отсутствие гарантии на доставку данных , что позволяет ему передвать данные с максимальным быстродействием.

Следовательно UDP применяется для :

- Передачи данных, чувствительных к задержке (голос, видео , данные режима реального времени)
- Передачи данных, нечувствительных к потерям пакетов
- Метод UDP-туннелирования , когда трафик приложений между сетями или устройствами помещается в UDP пакеты для ускоренного прохождения через сеть.
- Для передачи данных протокол UDP используют протоколы SNMP, RIP, DNS, RTP, SIP и другие

Слайд 5

Протокол UDP может передавать данные от одного приложения на одном оконечном устройстве другому приложению на другом оконечном устройстве, посредством портов .

Также UDP может проверять целостность пакета посредством контрольной суммы , но это необязательная функция.

Слайд 6

UDP получает сообщения от прикладного уровня, добавляет к ним поля номеров портов отправителя и получателя для демультиплексирования

приемной стороной, а также два других специальных поля и передает полученный сегмент сетевому уровню

Слайд 7

Протокол был разработан Дэвидом П. Ридом в 1980 году и официально определён в

RFC 768

С тех пор протокол официально не обновлялся

В 1995 году описывается использование псевдозаголовка IPv6 для расчета контрольной суммы UDP в RFC-1883

В 2004 году стандартизируется UDP-Lite (Lightweight User Datargram Protocol) в RFC 3828

Идентичный UDP он иначе использует поля контрольной суммы и длинны , когда они проверяют часть дейтаграммы

Номер протокола в заголовке IP = 136

Слайд 8

В 2008 году появляется RFC 5405 Unicast UDP Usage Guidelines for Application Designers , в котором описывается UDP-туннелирование, подходы по ограничению скоростей передачи данных и UDP-Lite

В 2011 году IANA обновляет перечень транспортных портов для UDP в RFC 6335

В 2013 году появляются RFC 6935 и RFC 6936 подробно описывающие вопросы нулевой контрольной суммы UDP и туннелированию UDP/UDP-Lite поверх IPv6

Слайд 9

Заголовок UDP состоит из 4 полей :

1. Порт отправителя
2. Порт получателя
3. Длина
4. Контрольная сумма

На каждое поле выделено 2 байта , следовательно , длина всего заголовка UDP составляет 8 байт

В IPv4 Порт отправителя и контрольная сумма необязательны к использованию

В IPv6 необязателен только порт отправителя

Слайд 10

В поле порт отправителя указывается номер порта, на который нужно прислать ответ. Если ответ не требуется значение поля принимается 0x0000

Слайд 11

В поле порт получателя указывается, порт на который должна поступить информация, содержащаяся в дейтаграмме

Слайд 12

В поле длина указывается длина дейтаграммы(заголовка и данных) в байтах

Минимальная длина равна длине заголовка — 8 байт.

Теоретически, максимальный размер поля — 65535 байт для UDP-датаграммы

(8 байт на заголовок и 65527 на данные).

Фактический предел для длины данных при использовании IPv4 — 65507 (помимо 8 байт на UDP-заголовок требуется ещё 20 на IP-заголовок).

Слайд 13

В поле контрольная сумма указывается значение, используемое для проверки заголовка и данных на наличие ошибок

Если рассчитанная контрольная сумма равна 0 – значение поля 0xFFFF

Если отправитель не рассчитал контрольную сумму – значение поля 0x0000

Если получатель обнаружил ошибку в контрольной сумме , дейтаграмма уничтожается .

Слайд 14

Рассчитаем контрольную сумму четырех 16-битных слов 0x398a 0xf802 0x14b2 0xc281

0x398a + 0xf802 = 0x318d

0x318d + 0x14b2 = 0x463f

0x463f + 0xc281 = 0x08c1

В конце выполняется инверсия всех битов получившегося числа

0x08c1 = 0000 1000 1100 0001 1111 0111 0011 1110 = 0xf73e

Примеры расчета контрольной суммы показаны в RFC 1071

Слайд 15

Порт — это программная структура, определяемая его номером — 16-битным целочисленным значением (от 0 до 65535) . Порт служит идентификатором программы на транспортном уровне.

UDP порты обеспечивают возможность для отправки и получения сообщений UDP. UDP порт функционирует как одиночная очередь сообщений для получения всех дейтаграмм, предназначенных для программы, указанной номером порта протокола. Это означает, что UDP-программы могут получать более одного сообщения за раз.

Все номера портов UDP, которые меньше чем 1024 - зарезервированы и зарегистрированы в Internet Assigned Numbers Authority (IANA).

Номера портов UDP и TCP не пересекаются.

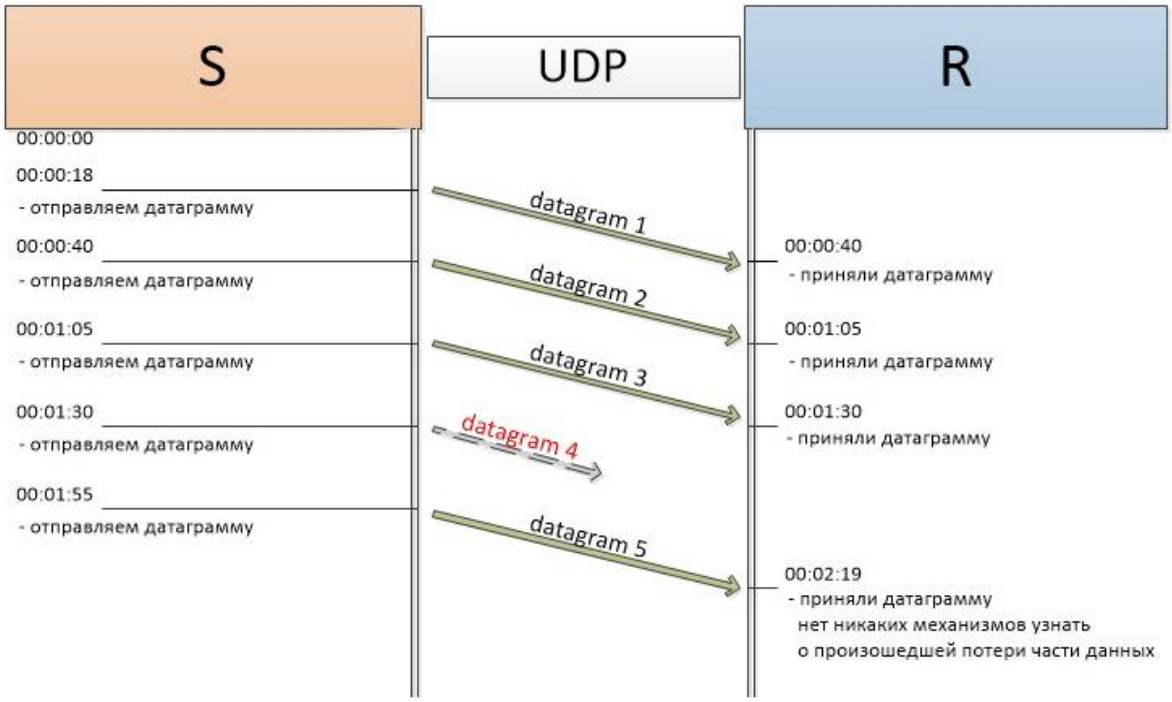
Слайд 16

Весь слайд таблица, на которой избражены известные протоколы пользовательского уровня и их UDP-порты

Слайд 17

Чувствительные ко времени приложения часто используют UDP (видеоданные), так как предпочтительнее сбросить пакеты, чем ждать задержавшиеся пакеты, что может оказаться невозможным в системах реального времени. Также потеря одного или нескольких кадров, при передаче видеоданных по UDP, не так критична, в отличии от передачи бинарных файлов, где потеря одно пакета может привести к искажению всего файла

Слайд 18

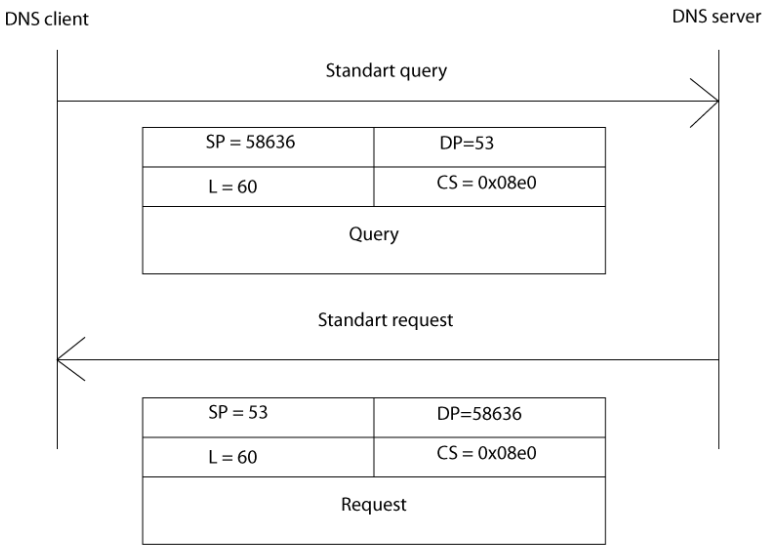


На данной картинке показан сценарий работы протокола UDP

Дейтаграммы передаются от порта S к порту R , при этом никаких сообщений о получении или не получении дейтаграммы, не отправляется .

4 Дейтаграмма теряется , повторно она не пересылается

Слайд 19



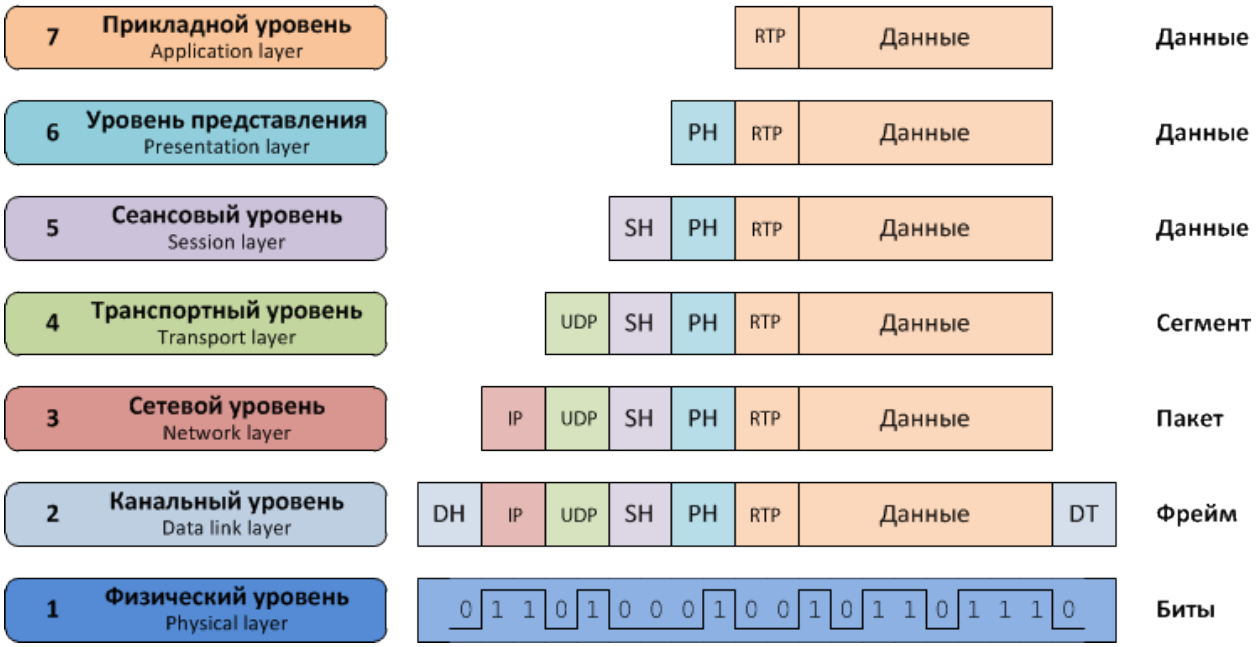
На данном слайде мы видим два UDP пакета : запрос и ответ

Такой сценарий используется, например, в протоколе DNS

Сначала отправляется запрос с порта 58636 на порт 53

Затем на порт 58636 отправляется ответ с порта 53

Слайд 20



На этом слайде мы видим инкапсуляцию, которую осуществляет протокол UDP

В данном примере данные протокола прикладного уровня RTP передаются через протокол транспортного уровня UDP

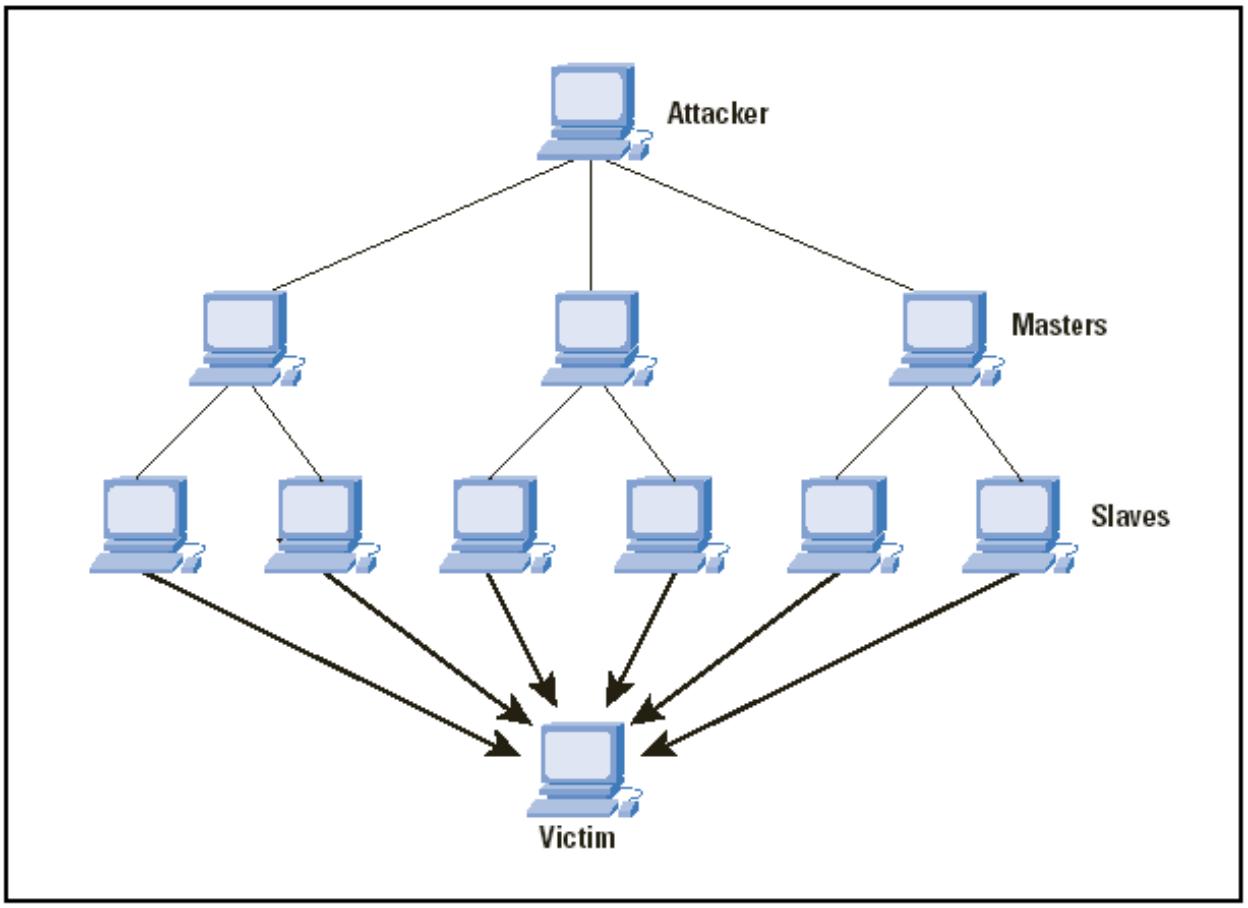
Для этого заголовок RTP и информация , помещается в поле данные заголовка UDP , создается UDP заголовок

Затем аналогично этому UDP передается через протокол сетевого уровня IP

И так далее

Слайд 21

Figure 4: A DDoS Attack



На этом слайде показана модель UDP флуда

Атакующий компьютер, захватывает другие компьютеры в сети и командует им посылать UDP-пакеты на определённые или случайные номера портов жертвы, который для каждого полученного пакета должен определить соответствующее приложение, убедиться в отсутствии его активности и отправить ответное ICMP-сообщение «адресат недоступен». Из-за большого числа пакетов жертва не успевает отвечать на все запросы (как полезные так и на запросы атакующего) и ответы приходят с задержкой или не приходят вовсе. В итоге сервис становится недоступен

Для защиты от атаки следует установить ограничение на количество обращений к открытым портам, а неиспользуемые порты закрыть средствами аппаратных или программных межсетевых экранов в ключевых точках сети.